基于多子群的社会群体优化算法 *

刘亚军，陈得宝[†]，邹　锋，李　峥，王苏霞

(淮北师范大学 物理与电子信息学院, 安徽 淮北 235000)

摘　要：社会群体优化算法(social group optimization，SGO)是一种基于社会群体学习而提出的一种新型优化算法。针对社会群体优化算法易于陷入局部最优问题，提出了一种多群社会群体学习算法（MPSGO）。本算法采用多子群学习方法，对算法两个阶段的个体学习方法进行改进，在维持群体收敛性能的前提下提高群体多样性，同时对部分个体中引入量子学习，使个体学习的有用信息得以增强；此外，每隔一定代数对子群进行随机重组，既能保证各子群个体充分进化，又维持了子群多样性。在设计算法的基础上，分析了其收敛性和多样性；通过与其他四种算法进行对比实验，验证了改进后算法性能更优。

关键词：社会群体优化算法；多子群；量子学习

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.12.0805

Multiple subgroups based social group optimization algorithm

Liu Yajun, Chen Debao[†], Zou Feng, Li Zheng, Wang Suxia

(*School of Physics & Electronic Information*, *Huaibei Normal University*, *Huaibei Anhui* 235000, *China*)

Abstract: Social group optimization (SGO) is a novel optimization algorithm which is based on social group learning. This paper proposed a multi-group social group learning algorithm (MPSGO) to solve the problem that the social group optimization algorithm was easy convergent to local optima. This algorithm adopted the multi-subgroup learning method, and the improvement of the individual learning method in the two stages of the algorithm. The diversity of the population was improved with maintaining the convergence of the population. At the same time, it introduced quantum-behaved learning method into MPSGO for parts of individuals to enhance the useful information of individual learning. In addition, the population was randomly regrouped to generate new subgroups after a certain generation. The diversity of subgroups wais maintained and the individuals in every subgroup were fully evolved. Based on the designing of algorithm, this paper analyzed the algorithm of convergence and diversity. Compared with the other 4 algorithms, it is proved that the performance of the improved algorithm is better.

Key words: social group optimization; multiple subgroups; quantum-behaved learning

0　引言

优化问题是现实生活中经常遇到的问题，如何设计算法快速准确地得到问题的最优解，是解决优化问题的关键一环。传统的优化算法往往依赖系统的精确数学模型和系统参数，对一些模型难以建立的系统，传统优化方法难以解决。自然计算方法模拟生物的自然或社会特性，其不依赖系统的精确模型，已得到广泛应用。经典的自然计算方法，如遗传算法(GA)[1]、粒子群算法(PSO)[2]、差分进化算法(DE)[3]、进化策略(ES)[4]等都已成功应用于函数优化、工程设计、自动控制、信号处理等领域。这些算法都利用个体间信息的交换或产生部分新信息实现对问题的优化，算法自身参数对算法性能影响较大，且最优化的参数往往难以确定，一定程度上影响了算法的应用。

近年来，为减少算法自身待确定参数数量，充分利用群体信息，研究者设计了一些新的优化算法。例如教学优化算法(TLBO)[5]，模拟班级教师的授课和学生学习过程，教师在班级分享自己的知识，提高班级的平均成绩，学生除向教师学习外，还向班级的其他学生学习，以进一步提高自己的成绩。此方法中由于引入教师的信息，算法收敛速度较快，但由于个体每次学习都保留最好的解，群体多样性丢失较快，算法易于陷入局

收稿日期：2017-12-13；**修回日期：**2018-01-29　**基金项目：**国家自然科学基金资助项目（61572224）；安徽省高校自然科学研究重大项目（KJ2015ZD36，KJ2016A639）；安徽省自然科学基金资助项目（1708085MF140）

作者简介：刘亚军（1993-），男，硕士研究生，主要研究方向为智能优化计算；陈得宝（1975-），男（通信作者），教授，博士，主要研究方向为智能计算、模式识别等（chendb_8@163.com）；邹锋（1978-），男，副教授，博士，主要研究方向为进化计算；李峥（1980-），男，副教授，硕士，主要研究方向为进化计算；王苏霞（1993-），女，硕士研究生，主要研究方向为智能优化计算.

部最优解。为进一步提高 TLBO 算法的性能，研究者提出了一些改进算法。例如基于邻域搜索的教学优化算法(NSTLBO)[6]在教学过程中引入个体的邻域信息，以增加个体学习方法的多样性，提高了 TLBO 算法的性能。精英教学算法(ETLBO)[6]在优化过程中保留多个精英个体，利用精英个体的引导作用，改变个体的跟踪行为，进一步提高算法的收敛精度。

为进一步提高算法的收敛速度，充分发挥个体的学习能力，Satapathy 等人[8]于 2016 年提出的一种基于社会群体学习的新型优化算法（SGO）。此算法操作简单，易于实现，在一些问题优化中取得了良好的效果[9]。此算法将个体的优化过程分为两个阶段，即提高阶段和获得阶段。在提高阶段，个体以群体当前最好解作为学习向导；在获得阶段，个体以群体中任意一个个体和当前代群体最好解作为指导进行学习，算法的收敛速度较快。一些典型数据集上的测试结果显示了 SGO 算法的有效性[8,9]。虽然 SGO 算法的收敛速度较快，但其至少存在两方面的不足：首先，SGO 算法在学习的两个阶段中均以群体最好个体作为向导，算法的收敛速度较快，但群体多样性容易丢失，导致算法易于陷入局部最优解；其次，由于算法学习方法单一，一旦陷入局部最优解，个体逃离局部最优解的能力不足。因此，在保持 SGO 算法收敛速度的前提下提高群体的多样性，成为提高算法整体性能的关键。本文在对个体更新方程改进的基础上，利用多群算法具有维持群体多样性的优点，结合量子学习强化所需解的能力，设计了一种多群社会优化算法。本文的贡献主要有三点：a)引入多子群学习提高群体的多样性；b)对每个子群中的个体，在获得阶段随机引入其他子群最优解，并将其与全局最优解和本群中其他个体信息相结合，增加子群个体学习方法的多样性；c)采用简单的子群生成方法，既保证了子群个体的充分进化，又达到子群间信息交换和改善子群多样性的目的。

1　SGO 算法

复杂问题的解决需要多方面因素的参与，当某个个体不足以解决复杂问题时，需要向周围或拥有此项能力的个体学习，以便提高自身解决问题的能力。SGO 算法是基于此种思想而提出的一种新型优化算法。SGO 算法实现过程主要包含初始化、提高、获得阶段三个步骤。描述如下：

1.1　种群初始化

随机初始化种群 P，如式(1)所示。

$$X_{i,j} = x_{\min} + r*(1,D)*(x_{\max} - x_{\min}) \tag{1}$$

其中：$i = 1,2,3,\ldots N$，$j = 1,2,3,\ldots D$；N 为种群的规模；D 为变量的维数；r 为 0 到 1 之间的随机数；$x_{\max}$ 与 $x_{\min}$ 分别为变量的上下限；P 为进化种群；X 为个体位置。

1.2　提高阶段

SGO 算法在此阶段采用的学习策略如式(2)所示。

$$Xnew_{i,j} = c*Xold_{i,j} + r*(gbest_j - Xold_{i,j}) \tag{2}$$

其中：c 为自我反省参数，其取值通常在 0~1 间，其经验值[8]为 0.2；r 为 0~1 间的随机数；$gbest_j$ 为当前代最优个体的第 j 维基因；$Xnew_{i,j}$ 与 $Xold_{i,j}$ 分别为第 i 个个体更新前后的第 j 维基因。

1.3　获得阶段

在获得阶段，每个个体从种群中随机选择一个个体作为学习对象，并结合当前代群体的最优个体信息进行学习。具体学习方法如式（3）（4）所示。

if $f(X_i)$ is better than $f(X_k)$

$$Xnew_{i,j} = Xold_{i,j} + r_1*(X_{i,j} - X_{k,j}) + r_2*(gbest_j - X_{i,j}) \tag{3}$$

else $$Xnew_{i,j} = Xold_{i,j} + r_1*(X_{k,j} - X_{i,j}) + r_2*(gbest_j - X_{i,j}) \tag{4}$$

其中：$X_{i,j}$ 与 $X_{k,j}$ 分别为个体 i 和个体 k 的第 j 维基因；$Xnew_{i,j}$ 与 $Xold_{i,j}$ 分别为第 i 个个体更新前后的第 j 维基因；$gbest_j$ 为当前代最优个体的第 j 维基因；r_1、r_2 都为 0~1 间的随机数。

2　多子群 SGO 算法（MPSGO）

在基本 SGO 算法的两个阶段中，个体均以当前代群体的最优个体为参考进行位置更新。实践表明，以最优个体引导进化能加快算法的收敛速度，但不利于维持群体的多样性，易导致算法陷入早熟收敛。多群操作是一种改善群体多样性的有效方法[10]，子群间信息的交互能为个体学习提供更多的信息。多群 SGO 对基本 SGO 算法的个体位置更新方法进行改进，充分利用子群的最优个体信息和整个群体的最优信息，引入量子学习方法，随机选择部分个体进行更新，以提高算法的收敛精度和增加群体多样性，减小算法陷入局部最优的可能。

2.1　提高阶段的改进

由式（2）可知，基本 SGO 算法的提高阶段，个体向群体的最优个体学习，目的是提高算法的收敛速度，而快速的收敛易导致群体多样性丢失，算法易于陷入局部最优。此外，从知识的学习难易程度看，个体更容易从其较近的邻域个体获得知识。基于这样的思想，在 MPSGO 算法中的提高阶段，以子群最优个体的信息对个体进行引导，利用就近获取知识方法使个体在其局部区域增长知识，增加个体学习知识的多样性。具体操作如下：首先将所有个体随机分成个体数量相等的若干个子群，再求取每个子群的最好个体，按照式（5）对各子群个体位置进行更新。

$$Xnew_{i,j} = c*Xold_{i,j} + r*(agbest(a,j) - Xold_{i,j}) \tag{5}$$

其中：c 的取值与基本 SGO 算法相同；r 为 0~1 间的随机数；a 表示第 a 个子群；$Xnew_{i,j}$ 与 $Xold_{i,j}$ 分别为第 i 个个体更新前后的第 j 维基因；$agbest(a,j)$ 为第 a 个子群中最好个体的第 j 维基因。

2.2　获得阶段的改进

由式（3）（4）可知，在 SGO 算法的获得阶段，个体向群

体中的其他个体及群体中的最好个体学习以获得新知识，但并未考虑个体学习能力的局限性和学习方法的多样性，不利于群体多样性的维持。基于这样的分析，在 MPSGO 算法的获得阶段，被选择用于学习的随机个体限制在子群内部，个体除了向子群内其他个体和种群最优个体学习外，还增加了个体向其他随机选择子群的最好个体学习，以提高个体学习方法的多样性。改进的获得阶段个体更新方程如式（6）（7）所示。

对某子群中的个体 i，在其所在的子群中随机选择个体 k 作为其学习的第一部分；在所有子群中，随机选择一个子群的最优解作为其学习的第二部分；种群的当前最优解作为其学习的第三部分，新的学习机制如下。

If X_i is better than X_k

$$Xnew_{i,j} = X_{i,j} + r_1 * \left(X_{i,j} - X_{k,j}\right) + r_2 * \left(gbest_j - X_{i,j}\right) + r_3 * \left(agbest(aa, j) - X_{i,j}\right) \tag{6}$$

else

$$Xnew_{i,j} = X_{i,j} + r_1 * \left(X_{k,j} - X_{i,j}\right) + r_2 * \left(gbest_j - X_{i,j}\right) + r_3 * \left(agbest(aa, j) - X_{i,j}\right) \tag{7}$$

其中：aa 为随机选择子群；$X_{i,j}$ 与 $X_{k,j}$ 分别为个体 i 和个体 k 的第 j 维基因；$gbest_j$ 为所有子群中最优个体的第 j 维基因；$agbest(aa, j)$ 为子群 aa 中最优个体的第 j 维基因；$Xnew_{i,j}$ 为 $X_{i,j}$ 为个体 i 更新前后第 j 维基因的位置。

式（6）（7）表明，在 MPSGO 算法的获得阶段，个体不仅从其所在子群中获得知识，同时也从全体子群的最优解和其他子群的最优解中获得知识，既兼顾了个体学习范围的局限性，又考虑到了个体学习方法的多样性。

2.3 量子学习策略

Grover 在 1996 年提出了一种高效的量子算法[11]，即量子搜索算法。随后，研究者把量子计算与机器学习结合起来，形成了量子学习[12~14]。量子算法特点就是利用量子态的干涉作用，使所需的结果增强，不需要的结果减弱，这样在检测时所需的结果就会以较高的概率出现[15]。MPSGO 算法中，在子群更新过程中引入量子学习，目的是提高较优解的出现概率。本方法中，以子群最优解（$agbest$）及个体与群体平均值之差（$X_i - aMean(a)$）作为个体学习的向导。为减少适应度函数的评价次数，本文仅对子群中随机选择的部分个体进行操作。方法如下。

$$Xnew_{i,j} = temp + ((-1) * ceil(0.2 + rand(1, j))) * b * \left(-\log(u)\right) \tag{8}$$

$$temp = \left(p. * X_{i,j} + m. * agbest(a, j)\right) / (p + m) \tag{9}$$

$$b = rand(1) * \left(aMeanX(a, j) - X_{i,j}\right) \tag{10}$$

其中：p,m,u 均为 0~1 间的随机数；$aMean(.)$ 为子群 a 中所有个体的位置平均解。

2.4 子群信息交换

子群间信息交换是保持各子群多样性的一个重要方法。由于子群规模往往较小，当迭代到一定代数时，子群中的个体易于趋同，导致子群多样性减小，不利于算法的收敛。此外，为给子群中的个体充分的进化时间，各子群必须进化一定的代数。为不增加算法的复杂性，在 MPSGO 算法中，采用一种简单的分群方法，当群体进化一定代数后，所有子群的个体随机重新组合，形成新的子群。具体操作方法为：如果进化代数能被给定的子群进化周期整除，则所有群体的个体重新混合，随机形成新的子群（If mod(*gen*,*T*)=0 then regroup the population）。*gen* 为当前的优化代数。*T* 为设定的整数值，*T* 越大越有利于子群充分进化，*T* 越小越有利于维持子群多样性。

2.5 算法收敛性分析

最小值优化问题如下：

$$\min\left\{ f(X) \mid \forall X \in \Omega, 0 < f(X) < +\infty \right\}$$

其中 Ω 是可行搜索空间且是非空有限集。

假设 1　搜索空间是范围为 Ω 的 *D* 维空间，其中有 N 个局部最优解，相对应的区域范围记是 $S_1, S_2, \ldots, S_N$，记 X*为局部区域 S*的最好解。

假设 2　存在 *N* 个子群，搜索区域为 $\Omega_1, \Omega_2, \ldots, \Omega_N$，每个子群可以同时搜索相对应的区域。

定理 1　如果前两个假设满足，则多子群算法比全群体算法具有更高找到全局最优解的概率。

证明　对于全群算法，令 $p\{X \in S^*\}$ 为找到最优解的概率，$p\{X \in S^*\}$ 定义如下：

$$p\{X \in S^*\} = \prod_{n=1}^{N} p\left\{X \in S_n\right\} = \prod_{n=1}^{N} \frac{S_n}{\Omega} \tag{11}$$

对多子群算法，令 $p_N\{X \in S^*\}$ 为找到最优解的概率，$p_N\{X \in S^*\}$ 定义如下：

$$p_N\{X \in S^*\} = \prod_{n=1}^{N} p\left\{X \in S_n\right\} = \prod_{n=1}^{N} \frac{S_n}{\Omega_n} \tag{12}$$

则

$$\frac{p_N\{X \in S^*\}}{p\{X \in S^*\}} = \frac{\prod_{n=1}^{N} \frac{S_n}{\Omega}}{\prod_{n=1}^{N} \frac{S_n}{\Omega_n}} = \prod_{n=1}^{N} \frac{\Omega}{\Omega_n} \tag{13}$$

因 $\Omega_n < \Omega$（n=1,2，…, N），则

$$p_N\{X \in S^*\} > p\{X \in S^*\} \tag{14}$$

由上述分析可得，多子群 MPSGO 算法找到最优解的概率优于单一群体的 SGO 算法。

对于 MPSGO 算法而言，群体中个体学习可以认为是变异操作算子(MO)，子代学习者与父代学习者的比较可以看做是选择操作算子(SO)[16]。在每次迭代中，MPSGO 算法优化过程依次由 MO 与 SO 实现，故利用随机泛函理论可以将一次迭代抽象为由 MO 和 SO 合成的随机映射。

定义 1　变异算子（MO）是一种以一定概率对个体向量的各维度分量进行重组变换的过程。它是解空间上的一种随机映射 T_M：$M \times \Omega \to \Omega^M$。在 MPSGO 中，这变异操作定义为

$$\begin{aligned} & p\{\omega \mid T_m(\omega, X_i) = newX_i\} \\ & = p\{newX_i = c*Xold_i \\ & + r*(agbest(a, j) - Xold_i)\} \end{aligned} \quad (15)$$

其中：ω 表示基本事件；其他参数含义与式（5）相同。

定义 2 选择算子（SO）是基于优胜劣汰的贪心策略思想来选择子代学习者的过程，它是解空间 Ω^M 与解空间 Ω 之间的映射关系，可以表达为 T_S 。$\Omega^M \to \Omega$ 。

对于 MPSGO 算法，其选择算子定义为

$$\begin{aligned} & T_S(X_i, newX_i) \\ & = Z \wedge Z \in \{X_i, newX_i\} \wedge f(Z) \\ & = \min\{f(X_i), f(newX_i)\} \end{aligned} \quad (16)$$

由于变异算子和选择算子都是随机生成的，所以产生的学习者也是随机的，使算法具有跳出局部最优的能力，不易陷入局部最优。在 MPSGO 的一次迭代中，相当于将映射 T 作用于当前群体 P，T 表达式如下所示：

$$T = T_S \cdot T_M : M \times \Omega \to \Omega \quad (17)$$

令 $X_i(t)$ 为第 t 次迭代时群体中的第 i 个个体，$X_i(t+1)$ 为第 $t+1$ 次迭代时群体中的第 i 个个体，那么

$$X_i(t+1) = T(\omega, X_i(t)) = T_S(T_M(\omega, X_i(t))) \quad (18)$$

在 T 作用下，MPSGO 每次迭代所产生的子代个体并不差于其相应的父代个体，因此，各代群体中最优学习者对应的适应值形成一个单调非增序列，即

$$\ldots f(X_{best}(t-1)) \geq f(X_{best}(t)) \geq f(X_{best}(t+1)) \ldots \quad (19)$$

引理 1[17] 设 d 是距离函数 $d(X_i, Y_j) = |f(X_i) - f(Y_j)|$，$\forall X_i, Y_j \in \Omega$ 定义一个从 $\Omega \times \Omega$ 到 R 的映射，那么，(Ω, d) 是一个完整的赋范空间。

定义 3 令 Ω 和 R 是两个赋范空间。定义 $X_t = T(\omega, X_{t-1})$，如存在一个随机变量 $K(\omega) < 1$ 使得

$$\begin{aligned} & p\{\omega \mid d(T(\omega, X_t), T(\omega, X_{t+1})) \\ & \leq K(\omega) d(T(\omega, X_{t-1}), T(\omega, X_t))\} = 1 \end{aligned} \quad (20)$$

称映射 $T: \Omega \to R$ 是一个随机压缩映射。

定理 2 MPSGO 的一次迭代所形成的随机映射 T 是一个随机压缩算子。

证明 据 MO 算子与 SO 算子的定义及式（18）可知，MPSGO 在每次迭代过程中，子代最好个体可能优于父代最好个体。因而存在一个随机变量 $K(\omega) < 1$，使得

$$\begin{aligned} & d(T(\omega, X_{t-1}), T(\omega, X_t)) = d(X_t, X_{t+1}) = |f(X_t) - f(X_{t+1})| \leq \\ & K(\omega) |f(X_{t-1}) - f(X_t)| = K(\omega) d(X_{t-1}, X_t) \end{aligned} \quad (21)$$

所以，

$$\begin{aligned} & \Omega' = \{\omega \mid d(T(\omega, X_t), T(\omega, X_{t+1})) \\ & \leq K(\omega) d(X_{t-1}, X_t)\} \subseteq \Omega \end{aligned} \quad (22)$$

因此，

$$p(\Omega') = 1 \quad (23)$$

在 MPSGO 中由 MO 和 SO 合成的映射 T 是一个随机压缩映射。

引理 2[17] 设 Ω 是一个巴拿赫空间。若 $T: \Omega \to \Omega$ 是一个压缩映射，则 T 在 Ω 内有唯一不动点。

定理 3 设 $T: M \times \Omega \to \Omega$ 是 MPSGO 中的压缩映射，那么，在 Ω 中 T 有唯一不动点，也即，MPSGO 是渐近收敛。

2.6 SGO 与 MPSGO 的多样性分析

本文选取不同类型的函数 F2、F4、F6 和 F8 来检验算法的多样性。测量多样性的方法很多，如基于所有个体平均汉明距离的方法；所有个体的位置与群体中心的距离方法以及熵方法[18]。为了简单直观分析 SGO 和 MPSGO 算法的多样性，本文采用基于平均距离来衡量群体多样性。为了公平比较 SGO 和 MPSGO 算法的多样性，两个个体之间的距离被归一化到[0,1]内。群体多样性定义如下：

定义 4 群体中个体 p_i 与 p_l 之间的距离为

$$H(p_i, p_l) = \sum_{j=1}^{D} |p_{i,j} - p_{l,j}| \quad (24)$$

定义 5 群体中个体 p_i 与 p_l 之间的归一化距离为

$$\begin{aligned} & H^*(p_i, p_l) = H(p_i, p_l) / \max\{H(p_m, p_n), \\ & m = 1, 2, \ldots N-1; n = m+1, m+2, \ldots, N\} \end{aligned} \quad (25)$$

定义 6 每次群体迭代中的平均距离为

$$\bar{H}(T) = \frac{2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} H^*(p_i, p_l)}{N*(N-1)} \quad (26)$$

在本次测试中，SGO 和 MPSGO 算法的参数设置如下：两算法的种群大小均为 50，以算法最大迭代次数作为终止条件，算法代数为 4 000，函数维数为 50，两种算法的多样性变化如图 1 所示。

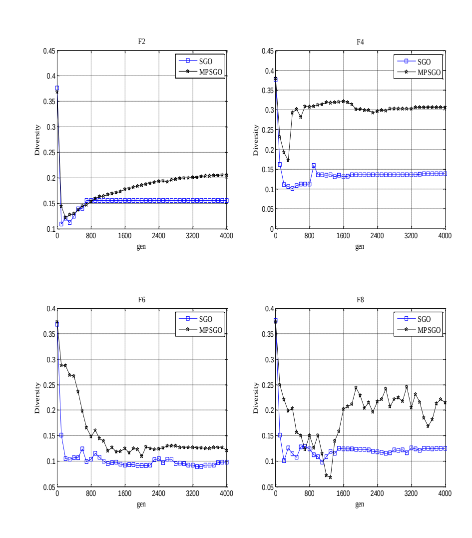


图 1 SGO 和 MPSGO 的多样性变化图

图 1 结果显示，对选取的四个函数，MPSGO 的多样性优于基本 SGO。F2 函数图像表明在初始化阶段，MPSGO 的多样性差于基本 SGO，但在中后期，MPSGO 的多样性优于基本 SGO；F8 函数图像显示，在迭代 1 200~1500 代时，MPSGO 的多样性差于基本 SGO，但在 1 500 代后，MPSGO 的多样性明显优于基本 SGO。F4 和 F6 函数图像表明，MPSGO 的多样性始终优于基本 SGO 算法。

2.7　改进算法流程

由以上分析，MPSGO 算法流程如下图 2 所示：

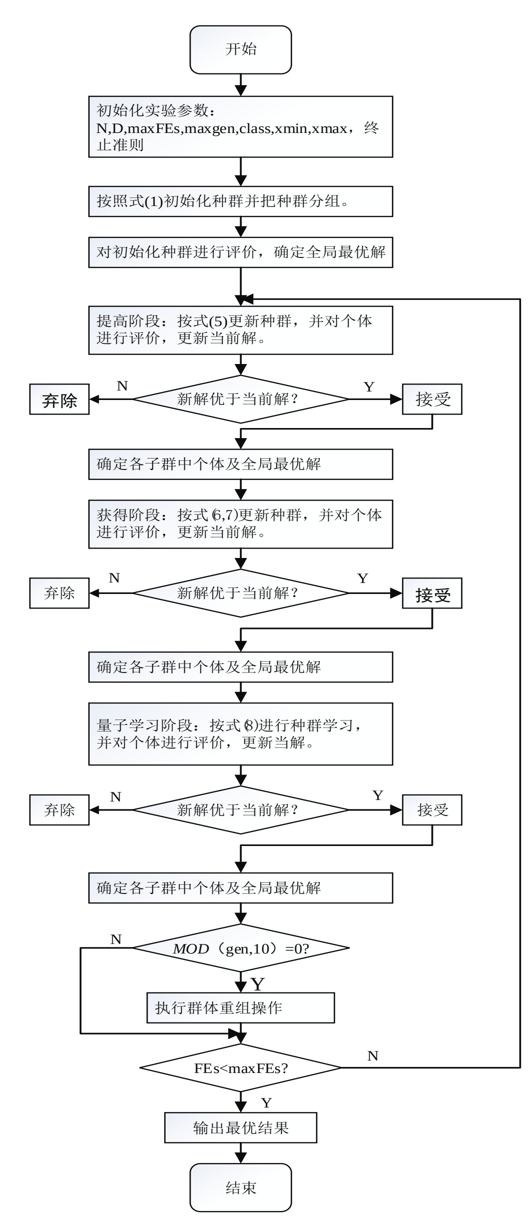


图 2　MPSGO 算法流程

3　仿真实验

为验证 MPSGO 算法的性能，本文对 8 个基准函数进行仿真实验，结果与 NSTLBO 算法[19]、ETLBO 算法[20]、PSOFDR 算法[21]、SGO 算法进行比较。

3.1　参数设置

本文所用的 8 个函数如表 1 所示。F1、F2 是单峰函数，F3、F4 为多峰函数，F5~F8 是旋转函数。所有算法的群体规模为 50，以函数值作为算法适应度值，最大函数评价次数(maxFES=5000*D)作为算法的停止准则，D 为变量的维数。子群数量为 10，c=0.2。

其他四种算法参数均来自相对应的参考文献，相关参数如下所示：

PSOFDR 算法[21]: $\omega\min = 0.4, \omega\max = 0.9, \varphi 1 = 1, \varphi 2 = 1, \varphi 3 = 2$;

NSTLBO 算法[19]: N（neighborhood Size）=3;

ETLBO 算法[20]:　elite size=2;

SGO 算法[8]:　c=0.2

表 1　测试函数

问题	函数	变量范围	最优值
F1	Zakharov	[−10,10]	0
F2	Rosenbrock	[−2.048,2.048]	0
F3	Ackley	[−32.768,32.768]	0
F4	Schwefel	[−500,500]	0
F5	Rotated Zakharov	[−10,10]	0
F6	Rotated Rosenbrock	[−2.048,2.048]	0
F7	Rotated Ackley	[−32.768,32.768]	0
F8	Rotated Schwefel	[−500,500]	0

3.2　实验结果及分析

为验证算法解决不同维优化问题的能力，分别对 30、50 维函数进行实验。每种算法独立运行 30 次，对其各种算法所得到的最优解（best）、平均值（mean）、标准差（std）进行统计分析，实验结果如表 2、3 所示。表 2、3 中粗体为最好结果。

从表 2 的结果可以看出，30 维函数条件下，对于函数 F1、F3、F5、F7，MPSGO 与 SGO 的性能相当，且三项性能都优于其他三种算法；对函数 F2、F4、F8，MPSGO 的性能优于其他算法；对函数 F2、F6，NSTLBO 的标准差优于其他四种算法；五种算法在函数 F3 的标准差相同。统计结果表明，MPSGO 的平均性能优于其他算法。

从表 3 的结果可以看出，在 50 维函数实验中，对函数 F1、F3、F5 和 F7，MPSGO 与 SGO 的性能相当，且三项性能都优于其他三种算法；对函数 F4 中，MPSGO 三种性能优于其他算法；在 F6、F8 实验结果中，MPSGO 的平均值优于其他算法；在 F8 函数实验中，NSTLBO 算法的标准差优于其他四种算法；在对 F3、F7 函数实验中，NSTLBO、MPSGO、SGO 三种算法标准差相同；在函数 F2 实验结果中，PSOFDR 的最优解、平均值优于其他算法。由表 2、3 统计分析可看出，MPSGO 的平均性能优于其他算法，尤其在高维度问题下，其优势较明显。

为了更直观地显示各算法平均适应度的变化过程，本文给出了不同算法对 50 维函数实验的优化过程中平均适应度变化，如图 3 所示。图 3 显示，对函数 F3、F4、F6、F7、F8，MPSGO 算法的收敛速度快于其他算法；MPSGO 在对函数 F2 实验前期收敛速度快于其他算法，后期收敛的精度差于 PSOFDR，但比 SGO 算法效果好。虽然在个别函数优化实验中，MPSGO 的收敛速度不是最快，但对于复杂函数，特别是旋转函数，MPSGO 表现出了良好的性能。

表 2　30 维函数测试结果

函数	性能指标	PSOFDR	NSTLBO	ETLBO	SGO	MPSGO
F1	best	1.99E-09	3.30E-73	1.93E-36	0.00E+00	0.00E+00
	mean	1.57E-08	1.03E-70	2.00E-34	0.00E+00	0.00E+00
	std	1.86E-08	1.65E-70	4.41E-34	0.00E+00	0.00E+00
F2	best	1.19E+01	2.40E+01	1.55E+01	2.79E+01	1.29E+01
	mean	1.41E+01	2.45E+01	1.72E+01	2.83E+01	1.34E+01
	std	2.41E+00	5.07E-01	1.19E+00	2.49E-01	4.26E-01
F3	best	7.11E-15	3.55E-15	3.55E-15	0.00E+00	0.00E+00
	mean	7.11E-15	3.55E-15	3.55E-15	0.00E+00	0.00E+00
	std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	best	2.82E+03	3.07E+03	3.59E+03	4.97E+03	1.78E+03
	mean	3.36E+03	4.43E+03	4.02E+03	5.61E+03	2.40E+03
	std	4.71E+02	8.11E+02	4.83E+02	6.06E+02	4.35E+02
F5	best	6.12E-10	2.60E-75	1.84E-38	0.00E+00	0.00E+00
	mean	5.55E-09	5.94E-71	7.80E-36	0.00E+00	0.00E+00
	std	7.19E-09	1.31E-70	1.04E-35	0.00E+00	0.00E+00
F6	best	2.45E+01	3.13E+01	1.62E+01	1.29E+02	1.17E+01
	mean	6.83E+01	5.69E+01	3.63E+01	4.16E+02	3.70E+01
	std	5.46E+01	2.28E+01	2.44E+01	2.29E+02	2.58E+01
F7	best	7.11E-15	3.55E-15	3.55E-15	0.00E+00	0.00E+00
	mean	1.86E-01	3.55E-15	3.55E-15	0.00E+00	0.00E+00
	std	4.16E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	best	1.84E+03	4.15E+03	3.16E+03	5.27E+03	2.01E+03
	mean	2.68E+03	4.92E+03	3.88E+03	6.47E+03	2.48E+03
	std	7.38E+02	7.98E+02	4.69E+02	1.09E+03	4.33E+02

表 3　50 维函数测试结果

函数	性能指标	PSOFDR	NSTLBO	ETLBO	SGO	MPSGO
F1	best	5.28E-04	1.08E-70	2.33E-31	0.00E+00	0.00E+00
	mean	2.64E-02	4.27E-68	2.96E-30	0.00E+00	0.00E+00
	std	3.29E-02	6.14E-68	3.86E-30	0.00E+00	0.00E+00
F2	best	2.67E+01	4.33E+01	3.63E+01	4.80E+01	2.75E+01
	mean	2.90E+01	4.38E+01	3.73E+01	4.85E+01	2.95E+01
	std	1.39E+00	3.33E-01	6.11E-01	3.46E-01	1.29E+00
F3	best	2.13E-14	3.55E-15	3.55E-15	0.00E+00	0.00E+00
	mean	3.29E-01	3.55E-15	3.55E-15	0.00E+00	0.00E+00
	std	7.36E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	best	4.46E+03	7.29E+03	6.40E+03	9.14E+03	4.05E+03
	mean	5.89E+03	8.01E+03	7.98E+03	1.01E+04	4.67E+03
	std	1.03E+03	7.00E+02	1.07E+03	7.00E+02	5.94E+02
F5	best	2.34E-03	3.75E-74	9.18E-34	0.00E+00	0.00E+00
	mean	1.40E-02	4.82E-68	6.95E-31	0.00E+00	0.00E+00
	std	1.33E-02	9.86E-68	1.46E-30	0.00E+00	0.00E+00
F6	best	5.12E+01	3.83E+01	4.93E+01	6.79E+02	2.23E+01
	mean	7.26E+01	1.07E+02	8.75E+01	1.66E+03	5.68E+01
	std	1.94E+01	4.83E+01	2.14E+01	1.24E+03	3.06E+01
F7	best	1.03E+00	3.55E-15	3.55E-15	0.00E+00	0.00E+00
	mean	1.32E+00	3.55E-15	4.26E-15	0.00E+00	0.00E+00
	std	2.07E-01	0.00E+00	1.59E-15	0.00E+00	0.00E+00
F8	best	5.68E+03	8.96E+03	8.38E+03	1.16E+04	3.94E+03
	mean	6.84E+03	9.81E+03	9.61E+03	1.26E+04	5.26E+03
	std	1.20E+03	8.15E+02	9.43E+02	8.88E+02	8.25E+02

为了进一步证明 MPSGO 算法与其他算法的差异，采用 t-test 方法对各种算法的结果进行测试。本文采用双边检验，显著水平为 0.05。在 30、50 维条件下，MPSGO 算法对于其他算法的 t 值和 p 值如表 4、5 所示。表 4、5 中，‘B’，‘S’，‘W’分别表示 MPSGO 算法性能优于、等于、劣于其他算法的数量。MPSGO 与其他算法之间的较好结果以粗体显示。由表 4、5 果可以看出，在采用双边检验，显著水平 0.05 情况下，MPSGO 算法性能优于大多数算法，解可以被接受。

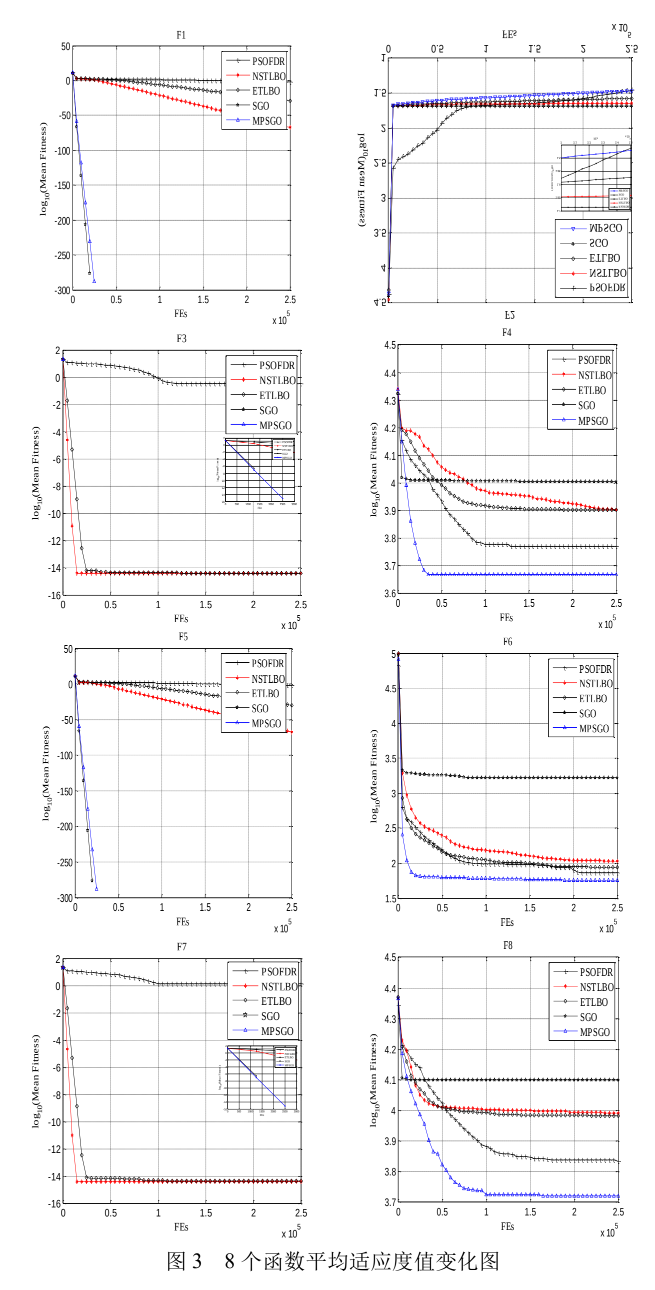


图 3　8 个函数平均适应度值变化图

4　结束语

本文针对 SGO 算法收敛精度低及群体多样性易丢失问题，提出了基于多子群的社会群体算法（MPSGO）。通过引入多子群、量子学习及子群信息交换方法，提高了 SGO 算法的整体性能。通过与 SGO 及其他三种算法比较实验，验证了改进方法的有效性。

表 4　各算法对 8 个基准函数为 30 维的 t-test 结果

函数	指标	PSOFDR	NSTLBO	ETLBO	SGO
F1	T	4.6258	3.4100	2.4821	0.0000
	P	0.0000	0.0012	0.0156	0.0000
F2	T	1.6905	92.2834	16.4651	165.6072
	P	0.0963	0.0000	0.0000	0.0000
F3	T	27.5191	13.7595	13.7595	0.0000
	P	0.0000	0.0000	0.0000	0.0000
F4	T	8.1838	12.0438	13.6443	23.5496
	P	0.0000	0.0000	0.0000	0.0000
F5	T	4.2326	2.4899	4.1208	0.0000
	P	0.0001	0.0157	0.0001	0.0000
F6	T	2.8442	3.1724	-0.1087	9.0180
	P	0.0061	0.0024	0.9141	0.0000
F7	T	2.4495	13.7595	13.7595	0.0000
	P	0.0174	0.0000	0.0000	0.0000
F8	T	1.2860	14.7259	12.0255	18.5730
	P	0.2053	0.0000	0.0000	0.0000
	B	6	8	7	4
	S	2	0	1	4
	W	0	0	0	0

表 5　各算法对 8 个基准函数为 50 维的 t-test 结果

函数	指标	PSOFDR	NSTLBO	ETLBO	SGO
F1	T	5.6831	4.9159	5.4326	0.0000
	P	0.0000	0.0000	0.0000	0.0000
F2	T	-1.9850	75.6295	38.6910	100.3924
	P	0.0519	0.0000	0.0000	0.0000
F3	T	3.1623	13.7595	13.7595	0.0000
	P	0.0025	0.0000	0.0000	0.0000
F4	T	7.2875	25.7462	19.1339	42.1701
	P	0.0000	0.0000	0.0000	0.0000
F5	T	7.4309	3.4557	3.3732	0.0000
	P	0.0000	0.0010	0.0013	0.0000
F6	T	3.0693	6.2383	5.8008	9.1123
	P	0.0033	0.0000	0.0000	0.0000
F7	T	45.0210	13.7595	18.9741	0.0000
	P	0.0000	0.0000	0.0000	0.0000
F8	T	7.6633	27.7548	24.5591	42.6334
	P	0.0000	0.0000	0.0000	0.0000
	B	7	8	8	4
	S	1	0	0	4
	W	0	0	0	0

参考文献：

[1] Giovanni L D, Pezzella F. An improved genetic algorithm for the distributed and flexible Job-Shop scheduling problem [J]. European Journal of Operational Research, 2010, 200 (2): 395-408.

[2] Alrashidi M R, El-Hawary M E. A survey of particle swarm optimization applications in electric power systems [M]. [S. l.] : IEEE Press, 2009: 913-918.

[3] Das S, Suganthan P N. Differential evolution: a survey of the state-of-the-art [J]. IEEE Trans on Evolutionary Computation, 2011, 15 (1): 4-31.

[4] Mallipeddi R, Mallipeddi S, Suganthan P N. Ensemble strategies with adaptive evolutionary programming [J]. Information Sciences, 2010, 180 (9): 1571-1581.

[5] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems [J]. Computer-Aided Design, 2011, 43 (3): 303-315.

[6] Rao R V, Rai D P, Balic J. Multi-objective optimization of machining and micro-machining processes using non-dominated sorting teaching–learning-based optimization algorithm [J]. Journal of Intelligent Manufacturing, 2016: 1-23.

[7] Rao R V, Patel V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems [J]. International Journal of Industrial Engineering Computations, 2012, 3 (4): 535-560.

[8] Satapathy S, Naik A. Social group optimization (SGO): a new population evolutionary optimization technique [J]. Complex & Intelligent Systems, 2016: 1-31.

[9] Naik A, Satapathy S C, Ashour A S, *et al.* Social group optimization for global optimization of multimodal functions and data clustering problems [J]. Neural Computing & Applications, 2016: 1-17.

[10] 廖锋，高兴宝．多群体差分演化算法及其应用 [J]. 计算机仿真, 2011, 28 (01): 230-233

[11] Luan L, Wang Z, Liu S. Progress of grover quantum search algorithm [J]. Energy Procedia, 2012, 16 (16): 1701-1706.

[12] Meyer D A. Quantum learning seminar lecture 1: introduction [EB/OL]. (2002) . http://math. ucsd. edu/~dmeyer/.

[13] Meyer D A. Quantum learning seminar lecture 2: grover's algorithm [EB/OL]. (2002) . http://math. ucsd. edu/~dmeyer/.

[14] Meyer D A. Quantum learning seminar lecture 3: single query learning [EB/OL]. (2002) . http://math. ucsd. edu/~dmeyer/.

[15] 张亮．采用量子进化算法学习的贝叶斯网络及其应用研究 [D]. 长沙: 湖南大学, 2011.

[16] Zou F, Chen D, Lu R, *et al.* Hierarchical multi-swarm cooperative teaching-learning-based optimization for global optimization [J]. Soft Computing, 2016: 1-22.

[17] 李敏强．遗传算法的基本理论与应用 [M]. 北京：科学出版社, 2002.

[18] Chen D B, Zhao C X. Particle swarm optimization with adaptive population size and its application [J]. Applied Soft Computing, 2009, 9 (1): 39-48.

[19] Wang L, Zou F, Hei X, *et al.* An improved teaching-learning-based optimization with neighborhood search for applications of ANN [J]. Neurocomputing, 2014, 143 (16): 231-247.

[20] Rao R V, Patel V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems [J]. International Journal of Industrial Engineering Computations, 2012, 3 (4): 535-560.

[21] Peram T, Veeramachaneni K, Mohan C K. Fitness-distance-ratio based particle swarm optimization [C]// Proc of IEEE Swarm Intelligence Symposium. 2012: 174-181.